소프트웨어 공학

Lecture #0: 소개

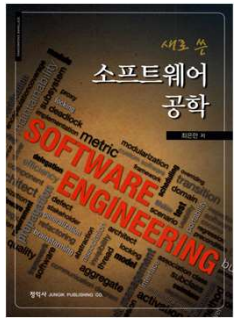
이수철
dakterlee@gmail.com

소개

- 담당교수
  - 이수철
  - dakterlee@gmail.com
- 예상하는 수강자
  - 프로그래밍 경험자
  - 실무 소프트웨어 개발 방법을 배우고 싶은 사람
  - 협업을 통하여 소프트웨어 개발을 경험하고 싶은 사람

교재

- 주교재
  - 최은만, 새로 쓴 소프트웨어 공학(정오표) (최은만 著) 정익사, 2015.
- 참고 문헌
  - Pressman, Software Engineering (8th edition), 2014.
  - Sommerville, Software Engineering (9th edition), 2010.



성적 평가 방법

- 중간, 기말, 레포트, 출석
  - 3,3,3,1
- 레포트
  - 개인별(UML작성), 팀별
- 출석 1/3이상 결석시 F
- 중간/기말 무단결시 F
- 레포트 표절 F
- 강의자료는 출력해서 빈칸넣기를 함
- 수업중 폰 사용 금지
- 수업전 교재는 꼭 읽어오기

강의 주제

- 소프트웨어 프로세스
- 계획
- 요구 분석
  - 구조적 분석
  - 사용 사례
- 설계
  - 설계 원리
  - 구조적 설계
  - 객체지향 분석 및 설계
    - Unified Modeling Language
  - 모듈 설계와 UI 설계
- 코딩
- 테스트
- 유지보수

주별학습내용

- 1. 강의소개, 소프트웨어공학 소개
- 2. 프로세스
- 3. 계획
- 4. 요구분석
- 5. 설계원리와 아키텍처
- 6. 객체지향 설계
- 7. 중간고사
- 8. 상세설계와 UI 설계
- 9. 코딩
- 10. 테스팅
- 11. 유지보수
- 12. 품질관리
- 13. 사례 1
- 14. 사례 2
- 15. 기말고사

강의사이트, 성적공지

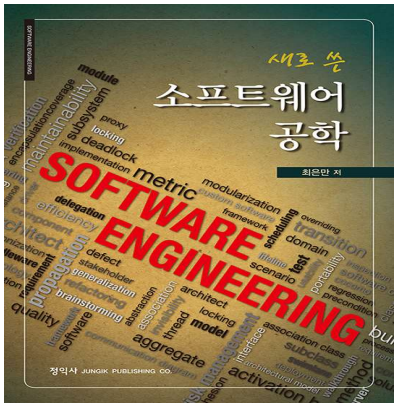
- Lab webpage? eclass? Facebook? Naver?
- http://cyber.sch.ac.kr/

쉬어가는 페이지

6차 개정판

소프트웨어 공학

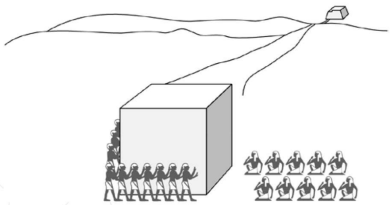
Lecture #1: 소개



학습 목표

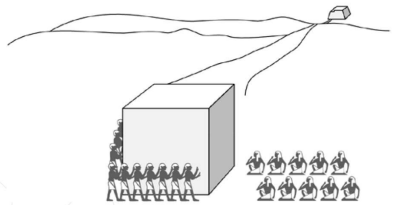
- 소프트웨어
- 소프트웨어 공학의 필요성
- 소프트웨어 공학이란?
- 소프트웨어 공학의 접근 방법
- 소프트웨어 공학 지식 체계

소프트웨어공학?



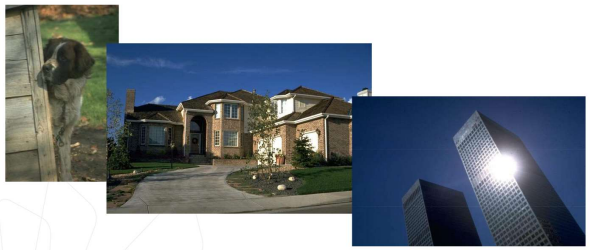
- 강에서 10K 떨어진 곳에 20명이 100일동안 피라미드 만들기

소프트웨어공학?



- Smart team
  - Go slow to go FAST
- Code-and-Fix 식 개발(X)
- Engineering 식 접근 방법(O)

소프트웨어공학?



- 규모에 따라
  - 계획, 설계, 만드는 방법, 전략이 달라짐

소프트웨어공학?

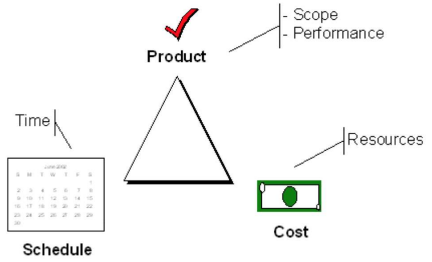
- 소프트웨어 개발에 엔지니어링 식의 접근 방법 적용
- 주먹구구가 아니라 효율적인 방법을 찾아보고 밝혀진 원리를 적용
  - 절차적인 프로그래밍의 설계 방법 – '구조적 방법'
  - 객체지향 프로그래밍의 설계 방법 – '객체지향 방법'
- 개발 전 단계에 대한 Best Practice 소개
  - 계획, 요구분석, 설계, 코딩, 테스트, 유지보수, 형상 관리 작업 방법
  - 관련 도구

소프트웨어 개발의 핵심 변수

- 소프트웨어 개발의 성패에는 다음과 같은 사항이 관련되어 있다.
  - 아이디어를 어떤 쓸모 있는 결과물로 바꾸는 과정(프로세스)
  - 제한된 시간과 자원 (인력, 자본)
  - 고객의 만족
  - 리스크를 관리
  - 개발 인력의 경험 및 팀워크(커뮤니케이션)

소프트웨어 프로젝트 트라이앵글

- 프로젝트는 성공적인 소프트웨어라는 결과를 내기 위하여 세 가지 요 소의 균형을 유지하여야 한다.



쉬어가는 페이지

- IT계열 기업의 연봉 순위를 알아보자 1~10위
  - http://officen.kr/officetalk/viewtalk.do?articleSeq=657

쉬어가는 페이지

소프트웨어와 우리 생활

- 의존성(dependability)

1.1 소프트웨어

- 소프트웨어
  - 프로그램 + 프로그램의 _____, _____, _____에 필요한 정보 일체
  - 개념적이고 _____(생산물의 구조가 코드 안에 숨어 있음)
- 소프트웨어의 특성
  - _____(_____)
  - 복잡성(Complexity)
  - _____(_____)
  - 복제 가능(Duplicability)

소프트웨어의 유형

소프트웨어 분류		특 징
응용 소프트웨어		• 비즈니스 업무 등 한 회사 또는 기관의 내부에서 사용하는 시스템 • 급여 시스템, 회계 시스템, 재고관리 시스템, 수강신청 등 학사업무 시스템, 은행 시스템
시스템 소프트웨어		• 운영체제, 장치 드라이버, 컴파일러, 코드 라이브러리
주문형 소프트웨어		• 특정 고객 또는 기업의 요구를 만족시키기 위하여 제작한 소프트웨어
패키지 소프트웨어		• 패키지화 하여 상업적으로 판매하는 소프트웨어 • 워드프로세서, 스프레드시트, 유통업체의 POS(Point of Sales) 시스템, 재정 분석, 주문 관리, 회계 관리 시스템
임베디드 소프트웨어		• 다른 시스템에 내장된 소프트웨어

소프트웨어와 시스템

- 시스템 : 필요한 기능을 실현시키기 위하여 관련 요소를 어떤 법칙에 따라 조합한 집합체
- 시스템의 성질
  - _____ – 교통시스템은 신호기, 신호체계, 도로망 등 여러 요소가 있고 이들 요소들은 원활한 교통 소통과 제어를 위해 밀접하게 연관
  - 기능적 분할 – 시스템은 규모가 작은 부속 시스템들로 나눌 수 있음
  - _____ – 시스템은 어떤 것이건 시스템과 주변 환경을 구분할 수 있는 경계가 있음. 이곳이 입력과 출력이 만나는 곳
  - 자동화 경계 – 시스템이 자동화된 부분과 수동 작업 부분을 나누는 경계

1.2 소프트웨어 공학의 필요성

- 소프트웨어 공학은 소프트웨어에 있는 심각한 직접적인 손해 또는 간접적인 손해가 따를 수 있는 문제를 해결
- 소프트웨어 제품은 _____의 문제를 해결하기 위해 구축, 비즈니스를 운영하기 위하여 사용(여기에서 비즈니스는 재고 관리, 재정 회계, 의료 정보, 교통 제어 관리 등 광범위한 의미)
- 소프트웨어가 제대로 작동하지 않으면 _____이 크고 사용자가 불편을 겪음

고비용

- LOC(Lines of Code) : 소프트웨어의 규모를 측정하는 데 가장 널리 사용
- MM(Man-Month) : 소프트웨어 개발에 드는 비용
- _____ : MM당 생산하는 프로그램의 LOC
  - 5만줄의 프로그램 – 4천만원 내지 1억 2천 정도의 비용
- _____ 현상

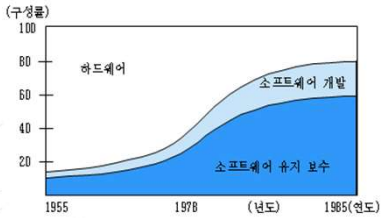
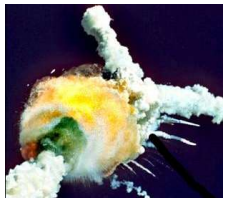


그림1.3 하드웨어와 소프트웨어 비용 구성 추이

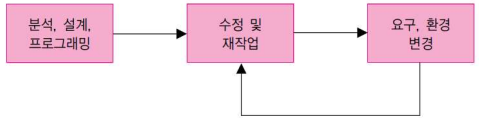
개발 지연과 낮은 신뢰도

- 계획에서 벗어난 컴퓨터 관련 개발 프로젝트
  - 600여 회사를 조사하였더니 35% 이상
- 예상대로 작동하지 않는 사례
  - 방위산업 보고(70% 이상)
- 다른 요소(하드웨어)와 다름
  - 노후화에 의한 물리적 특성의 변화에 의한 것이 아님
  - _____, _____과정에 유입된 오류에 의한 것



유지보수와 재작업

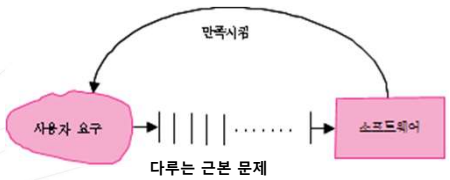
- 유지보수가 필요한 이유
  - 시스템에 남아 있는 _____가 있기 때문에
  - 소프트웨어는 _____가 흔하기 때문에 수정이 필요
- 소프트웨어 개발의 문제점
  - 의도하는 바가 잘 드러나지 않음 >> _____를 파악하기 어려움
  - 개발을 진행하면서 _____가 변경되고 재작업이 필요



1.3 소프트웨어 공학이란?

- 소프트웨어 공학 : 소프트웨어의 개발과 운영, 유지보수, 소멸에 대한 _____ 접근 방법
  - _____ 접근 : 소프트웨어 개발에 사용되는 방법이 일회성이 아닌 반복 사용이 가능함
- 고강도의 소프트웨어는 _____의 문제를 해결

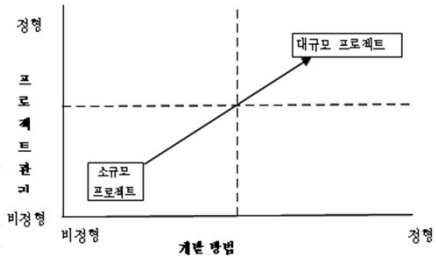
  >> ____의 요구를 만족시키기 위해 소프트웨어를 ____적으로 개발



다루는 근본 문제

규모

- 수만 줄의 소프트웨어를 개발할 때는 수백 줄의 프로그램을 개발하는 데 사용하는 방법과는 다른 방법을 적용
- _____식 접근 방법 - 방법, 절차, 도구 사용



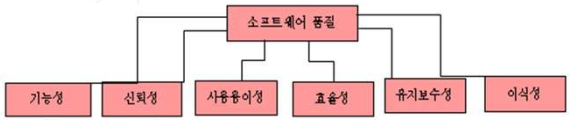
규모 문제

품질과 생산성

- 엔지니어링 작업에서는 _____, _____, _____과 같은 변수가 중요
- _____
  - Man-Month로 측정
- _____
  - 짧은 time-to-market
- _____

품질

생산성



품질과 생산성

- 품질을 나타내는 속성
  - _____(_____)
    - 소프트웨어가 사용될 때 원래 정한 또는 내재된 요구를 만족시키는 기능을 제공하는 능력
  - 신뢰성(reliability)
    - 소프트웨어가 정해진 수준의 성능을 유지할 수 있는 능력
  - _____(_____)
    - 쉽게 이해되고 배울 수 있고 사용될 수 있는 능력
  - 효율성(efficiency)
    - 사용되는 자원의 양에 따라 적절한 성능을 제공할 수 있는 능력
  - _____(_____)
    - 정정, 개선, 적응시킬 목적으로 수정될 수 있는 능력
  - 이식성(portability)
    - 별도의 작동이나 수단 없이 다양한 환경에서 적응될 수 있는 능력

일관성과 재현성

- 일관성
  - 프로젝트의 결과를 어느 정도 정확하게 예측가능
  - 더 높은 품질의 제품을 생산
- 프로세스의 _____가 필요
  - ISO 9001
  - CMM(Capability Maturity Model)
- 재현성
  - 개발하는 시스템 마다 높은 _____과 _____을 갖도록 만드는 것
  - 개발 능력, 결과의 재현성

변경

- 오늘날 비즈니스 환경 변화는 매우 _____
  - 소프트웨어 또한 시장에 따라 계속 진화하고 변경됨
  - 소프트웨어는 변경을 어렵게 하는 물리적인 부분이 없음
- 변경을 _____하고 _____하는 것이 또 하나의 과제
  - 변경은 소프트웨어 공학의 중요한 성장 요인

소프트웨어 공학의 접근 방법

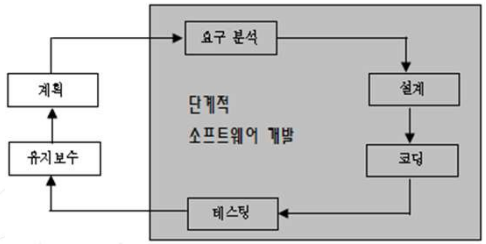
- 프로젝트를 수행하는 동안 얻은 품질과 생산성은 여러 가지 요인에 좌우됨
- 품질을 좌우하는 세가지 : _____, _____, _____
  - 프로젝트 _____

소프트웨어 공학의 접근 방법

- 소프트웨어를 개발하는 _____를 소프트웨어와 분리
- 소프트웨어 공학은 소프트웨어 제작과정에 집중
  - 알고리즘, 운영 체제, 데이터베이스 등은 소프트웨어 제품 자체에 초점
- 소프트웨어 엔지니어링 작업의 종류
  - 소프트웨어 개발 _____ – 시스템에 대한 비전과 개념을 목표로 하는 컴퓨터 환경에 실행되는 소프트웨어로 바꾸는 작업
  - _____ (Software Quality Assurance) – 개발작업이 적절히 수행되었는지 확인
  - 프로젝트 _____ – 개발과 품질보증 작업을 관리하고 감독

단계적 개발 프로세스

- 단계절 개발 프로세스를 따르는 이유
  - 소프트웨어의 문제를 나눠 여러 개발 단계에서 다른 ____을 다루기 때문
  - 개발하는 동안 정해진 시점에 품질과 진행을 체크할 수 있음



단계적 개발 프로세스

- 요구분석 : 소프트웨어 시스템이 풀어야 할 문제를 이해하기 위한 작업
  - 시스템이 목표를 어떻게 성취? vs _____?
  - _____ : 문제와 그 배경을 잘 이해하고 개발할 시스템의 요구 찾기
  - _____ : 요구명세서(requirement specification)
    - 시스템의 기능 이외에도 설계에 영향을 주는 모든 요인을 문서에 기술
- 설계 : 요구문서에 기술된 문제의 _____을 계획
  - 요구를 어떻게 만족시킬 것인지 추구
  - _____ 설계 : 시스템을 여러 컴포넌트의 집합체로 보고 각 컴포넌트들이 요청한 결과를 위하여 어떻게 상호작용 하는지에 초점
  - _____ 설계 : 각 모듈의 내부 논리를 작성

단계적 개발 프로세스

- 코딩
  - 시스템 설계를 프로그래밍 언어로 변환
  - 코딩 작업 중에는 읽기 쉽고 이해하기 쉬운 프로그램이 되어야함
  - _____과 _____ 을 추구
- _____ : 소프트웨어의 결함을 찾아냄
  - 소프트웨어 개발 단계에서 사용되는 중요한 품질 제어 수단
  - 프로그램에 포함된 요구, 설계, 코딩 오류를 밝힘
  - 단위 테스팅(unit testing) : 모듈이나 컴포넌트를 개별적으로 시험
  - _____(_____) : 모듈 사이의 연결을 시험
  - 인수 시험(acceptance testing) : 시스템이 잘 실행되는지 고객에게 데모

일반적인 개발 단계

표 1.3 개발 프로세스 각 단계

단계	초점	주요작업과 기술	결과물
분석	● 시스템을 위하여 무엇을 만들 것인가?	1. 분석 전략 수립(3장) 2. 요구 결정(3장) 3. 사용 사례 분석(4장) 4. 구조적 모델링(6장) 5. 동적 모델링(6장)	_____
설계	● 시스템을 어떻게 구축할 것인가?	1. 설계 전략 수립(7장) 2. 아키텍처 설계(5장) 3. 인터페이스 설계(7장) 4. 프로그램 설계 5. 데이터베이스, 파일 설계(7장)	설계 명세서
구현	● _____과 _____	1. 프로그래밍(8장) 2. 단위 테스팅(9장) 3. 시스템 안정화 및 유지보수(10장)	새 시스템, 유지보수 계획
테스팅	● 시스템이 요구에 맞게 실행되나?	1. 통합 테스팅(9장) 2. 시스템 테스팅(9장) 3. 인수 테스팅(9장) 4. 시스템의 설치(9장) 5. 프로젝트 관리 계획	테스팅 결과 보고서

품질 보증

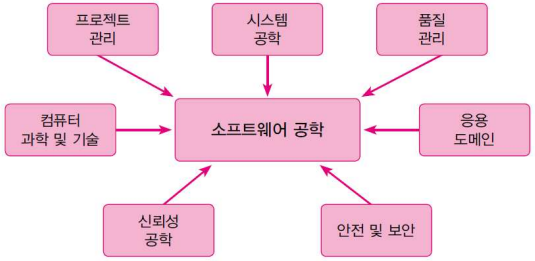
- 품질 보증 : 개발되고 있는 소프트웨어가 _____ 와 품질 수준을 만족시킬 것이라는 것을 보장하는 작업
- _____ (verification)
  - 개발작업이 프로젝트를 위해 선택된 프로세스와 방법에 맞게 수행되었는지 체크
  - 요구된 소프트웨어 결과물이 품질 수준에 맞게 생산되었는지 검사
- _____(validation) : 개발 프로세스에 의하여 생성된 결과물의 정확성을 체크
  - 정적 _____(static validation) : 소프트웨어를 실행시키지 않고 결과물의 정확성을 체크
  - 동적 _____(dynamic validation) : 소프트웨어를 실행시켜 잘 작동하는지 확인
- 테스팅
  - 동적 _____ 작업, 테스트 결과가 예상되는 결과와 일치하는지 체크

프로젝트 관리

- 프로세스와 관련된 이슈를 적절히 관리
  - 작업 과정에 자원을 어떤 작업에 할당할 것인지 기술
  - _____ 은 프로젝트의 개발 프로세스를 모니터링하고 제어하는데 사용되는 기준이 됨
- 프로세스 관리
  - 객관적인 데이터가 필요. 소프트웨어 메트릭이 사용
  - _____ 메트릭 : 개발한 프로덕트, 소프트웨어 자체의 특성을 계량화
  - _____ 메트릭 : 소프트웨어 개발에 사용된 프로세스의 생산성을 계량화

1.5 소프트웨어 공학 지식 체계

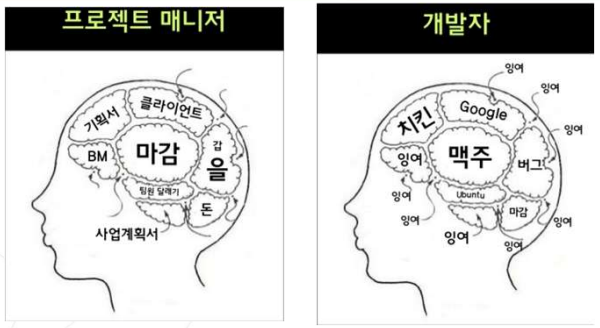
- 다른 분야와의 관계



SWEBOK

그림 1.15 ▶ SWEBOK 주요 지식 영역

쉬어가는 페이지



프로젝트(다음주까지 할 일)

- 프로젝트 팀 구성
  - 주제부터 모든 것 스스로 정함
- 4명으로 구성
  - 실제 프로젝트와 같이 팀구성(리더, writer, 엔지니어, ...)
  - 정확한 역할 분담과 팀 스피릿 발휘
- 단계별 결과
  - 제공한 템플릿을 이용한 결과물 생성
- 평가 요소
  - 내용, 완성도, 스타일(포맷)
- 웹
  - http://cyber.sch.ac.kr/

Questions?